1. Introduction to Robot Framework

Robot Framework is an open-source **test automation framework** mainly used for:

- Acceptance testing
- Regression testing
- RPA (Robotic Process Automation)

Key Features

- Keyword-driven testing
- Easy-to-read, tabular syntax
- Supports Selenium, API, Database, Appium, etc.
- Platform and language independent (written in Python)

File Types

- `.robot` → test cases
- `.resource` → reusable keywords
- `.py` → custom libraries

---

2. Environment Setup & Required Packages

Step 1: Install Python

Check installation:

```
python --version
```

Step 2: Install Robot Framework

```
pip install robotframework
```

Verify:

```
robot --version
```

Step 3: Install Selenium Library

```
pip install robotframework-seleniumlibrary
```

Step 4: Install WebDriver

Use ChromeDriver / EdgeDriver
(or use WebDriverManager manually if preferred)

---

3. Robot Framework – RIDE Tool

What is RIDE?

RIDE (Robot IDE) is a GUI tool to:

- Write test cases
- Manage keywords
- Execute tests

Install RIDE

```
pip install robotframework-ride
```

Launch RIDE:

```
ride.py
```

---

4. Writing a Simple Test Case (Using RIDE / Editor)

Example: Simple Login Test

```
*** Settings ***
Library    SeleniumLibrary

*** Test Cases ***
Open Google
    Open Browser    https://www.google.com    chrome
    Title Should Be    Google
    Close Browser
```

---

5. Keyword Driven Test Cases

What are Keywords?

Keywords are reusable actions (like functions).

Example: User Defined Keyword

```
*** Keywords ***
Open Application
    Open Browser    https://example.com    chrome
    Maximize Browser Window
```

Using Keyword in Test Case

```
*** Test Cases ***
Launch App Test
    Open Application
    Close Browser
```

---

6. Data Driven Test Cases

Example Using Test Template

```
*** Settings ***
Test Template    Login Test

*** Test Cases ***
Valid Login      admin    admin123
Invalid Login    user     wrongpass

*** Keywords ***
Login Test
    [Arguments]    ${username}    ${password}
    Input Text    id=user    ${username}
    Input Text    id=pass    ${password}
    Click Button  id=login
```

---

7. Executing Test Cases

Run All Tests

```
robot tests/
```

Run Specific File

```
robot login.robot
```

Generate Reports

- `report.html`
- `log.html`
- `output.xml`

---

8. Command Line Options

```
robot -d results tests/
robot -i smoke tests/
robot -e regression tests/
```

Option	Description
`-d`	Output directory
`-i`	Include tag
`-e`	Exclude tag

---

9. Test Setup and Teardown

Example

```
*** Settings ***
Test Setup     Open Browser To App
Test Teardown  Close Browser

*** Keywords ***
Open Browser To App
    Open Browser    https://example.com    chrome
```

---

10. Working with Browsers Using Selenium Library

Common Browser Keywords

```
Open Browser
Close Browser
Maximize Browser Window
Go To
```

---

11. Working with UI Elements

Text Box

```
Input Text    id=username    admin
```

Radio Button

```
Select Radio Button    gender    male
```

Check Box

```
Select Checkbox    id=agree
```

```
Unselect Checkbox  id=agree
```

Drop Down

```
Select From List By Label    id=country    India
Select From List By Index   id=country    2
```

---

12. Built-in Libraries in Robot Framework

BuiltIn

```
Log    Hello World
Set Variable    ${x}    10
```

Collections

```
Create List    a    b    c
```

String

```
Convert To Upper Case    robot
```

OperatingSystem

```
File Should Exist    test.txt
```

DateTime

```
Get Current Date
```

---

13. Summary

- Robot Framework uses **keywords instead of code**
- Easy to learn and readable
- Supports **keyword-driven & data-driven** testing
- Powerful integration with **Selenium**
- Ideal for **manual testers moving into automation**
